



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/007,084	12/04/2001	Jaroslav Gericic	YOR920010463US(8728-525)	5211
46069	7590	10/19/2006		EXAMINER
F. CHAU & ASSOCIATES, LLC 130 WOODBURY ROAD WOODBURY, NY 11797			SAIN, GAUTAM	
			ART UNIT	PAPER NUMBER
			2176	

DATE MAILED: 10/19/2006

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**MAILED**

OCT 19 2006

**Technology Center 2100**

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/007,084

Filing Date: December 04, 2001

Appellant(s): GERGIC ET AL.

Frank V. DeRosa, Reg. No. 43584  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 7/24/2006 appealing from the Office action  
mailed 4/7/2006.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

6578000	Dodrill et al	4-2000
6490564	Dodrill et al	2-2000
6424945	Sorsa	12-1999

"Voice Extensible Markup Language (VoiceXML) version 1.0", WWW Consortium, May 5, 2000.

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

**Claims 1-8, 10-12 and 33-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Voice extensible Markup Language (voiceXML) version 1.0 (May 5, 2000; WWW Consortium)(hereinafter "Vxml"), in view of Dodrill et al (US 6578000, filed Apr 28, 2000).**

**Regarding independent claim 1, Vxml teaches creating one or more reusable VoiceXML dialog components. For example, Vxml discloses creating a reusable library of dialogs shared among many applications with sessions where the user starts to interact with a voice interpreter context (see vxml, section 3, 3.1, 3.2), wherein creating reusable components (section 5, subdialogs heading).**

Vxml teaches creating an associated parameter object for each of the reusable voicexml dialog components. For example, Vxml discloses field terms that invoke objects with various parameters (sec 6.1, Field items heading).

Vxml does not expressly teach, but Dodrill teaches creating a Voice XML document comprising code for invoking a reusable voicexml dialog component and code for configuring the invoked reusable voicexml dialog component using an associated parameter object. Vxml discloses dynamically generated HTML forms developed XML-based voice web application using previously defined parameters (col 3, line 60 – col 4, line 38).

Vxml does not expressly teach *wherein the step of creating a reusable voiceXML dialog component comprises creating a re-entrant reusable voicexml dialog component that allow reusable voicexml dialog component to be one of initiated, interrupted, inspected or resumed with a partially filled result object or state object*. However Vxml does teach creating a reusable library of dialogs shared among many applications with sessions where the user starts to interact with a voice interpreter context (see vxml, section 3, 3.1, 3.2) and voicexml elements for “<initial>,” which declares initial logic upon entry into a (mixed-initiative) form (see vxml, section 4). Additionally, The Dodrill patent reference discloses development of a voice enabled web application using a browser, where the server parses the existing XML document that defines a voice application operation, inserts selected tag data into fields of a form in order to create or modify voice application operations in the entry fields, where the application server stores the XML document for later execution (col 7, lines 45-67). Examiner interprets

that creation or modification of a voice enabled XML document that is stored for modification is equivalent to the claimed creating because building components is the same thing as creating them.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Vxml with Dodrill to include development of a voice enabled web application using a browser, where the server parses the existing XML document that defines a voice application operation, inserts selected tag data into fields of a form in order to create or modify voice application operations in the entry fields, where the application server stores the XML document for later execution as taught by Dodrill, providing the benefit of a non-conventional application for voice enabled web applications using a browser, enabling users to input or modify application parameters for the XML document in the form (col 3, lines 47- col 4, line 1).

**Regarding claim 2**, Vxml does not expressly teach but Dodrill teaches populating object with appropriate parameter values. Dodrill discloses input application parameters into the entry fields (col 4, lines 20-21).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Vxml to include inputting application parameters into entry fields as taught by Dodrill, providing the benefit of a browser-based arrangement for developing voice enabled web applications using extensible markup language documents (Title).

**Regarding claim 3**, Vxml teaches invoking a reusable voicexml dialog component comprises a subdialog elements. Vxml discloses creating reusable components (section 5, Subdialog heading).

**Regarding claim 4**, Vxml teaches parameter object comprises a ECMAScript parameter object. Vxml discloses specifying a block of ECMAScript (col 4, lines 20-21).

**Regarding claim 5**, Vxml teaches building a library of reusable voicexml documents. Vxml discloses creating a reusable library in voicexml document where dialogs are in the document (sec 3; fig 2, 3 – vxml documents – Application root, main document).

**Regarding claim 6**, Vxml teaches building a reusable voicexml dialog module comprising a standard set of reusable voicexml dialog components. Vxml discloses creating a reusable library of dialogs (section 3.1, 3.2).

**Regarding claim 7**, Vxml teaches default prompts, object-specific resources, constructors that combine default and application specific parameters, and a method for manipulating parameter content, and a combination thereof. Vxml discloses a platform-specific document message to prompt users (section 5, Main Document example) containing resources relevant to object (section 14.5 – table, see archive object) with attributes have either default or specified fields by the last user input)(section 15, Filled heading and a retrieval of the content (section 12.1, see fetchint). Combining these elements would have been obvious to one of ordinary skill in the art at the time of the invention because the vxml reference teaches all of the elements providing the benefit of reusable components to take advantages of web-based development and content delivery to interactive voice response application (vxml, Abstract section).

**Regarding claim 8**, Vxml teaches code for dynamically compiling a grammar.

Vxml teaches fields using explicit grammars which are executed at runtime (section 14.1.2).

**Regarding claim 10**, Vxml teaches re-entrant objects are used for mixed initiative. Vxml discloses a mixed-initiative conversation (Abstract, 6.2.2, <initial>).

**Regarding claim 11**, Vxml teaches code for calling application-specific objects comprising interaction objects and service objects. Vxml discloses interacting with custom extension and "platform specific objects" (section 4 table, <object>; section 14.5).

**Regarding claim 12**, Vxml teaches a subdialog element and wherein the code for calling a service object comprises an object element. Vxml discloses a form item includes <subdialog> and form for weather information service calls objects within it (section 6.4).

**Regarding independent claim 33**, Vxml does not expressly teach, but Dodrill teaches a voicexml page generation engine for dynamically building a voicexml page. Dodrill discloses dynamically generated HTML form which is developed using XML-based voice web application using previously defined parameters)(col 3, line 60 – col 4, line 38).

Vxml does not expressly teach, but Dodrill teaches a first database comprising one or more server-side reusable voicexml dialog components that are accessible by the voicexml page generation engine for generating an intermediate voicexml page. Dodrill's discloses where a user can modify an application user the HTML forms and

then submit the HTML forms to the application server with generates corresponding XML document stored for later execution (col 8, lines 11-28, col 7, lines 40-67; fig 2). The examiner interprets Dodrill's storing of the XML document for later execution as equivalent to the database comprising reusable voicexml dialog components because the storing xml documents with voice enabled data must be stored in a data store for performing the storing step. Additionally, the reference Vxml does discloses Vxml subdialogs that permit the reuse of common dialog or build libraries of reusable applications for VXML applications. Since the vxml subdialogs are stored for reuse, they must be stored in some form of a data store.

Vxml does not expressly teach, but Dodrill teaches a second database comprising backend data that is accessible by the voicexml page generator to insert data in the intermediate voicexml page to generate a voicexml page that is served to a requesting client (i.e., inserting parameters into entry field for display of the form by voice enabled browser (Title) with XML tags)(col 10, lines 17-37; fig 5A).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Vxml to include developing XML-based voice web application using previously defined parameters and dynamically generating web pages with SQL database interfaces for inserting values into entry field for display of the form by voice enabled browser the as taught by Dodrill, providing the benefit of a browser-based arrangement for developing voice enabled web applications using extensible markup language documents (Title).

**Regarding claim 34**, vxml teaches beans and the voiceXML page generation engine comprises a JSP engine. For example, Vxml discloses the user of Java which can be used with voiceXML grammar elements (Vxml, Appendix D) which is equivalent to the claimed limitation because Java was well known in the art at the time of the invention and using Java beans was also well known in the art at the time of the invention to develop web server based reusable components for web applications to server data in forms, and this is evidenced by the Java Speech Grammar Format that defines a grammar for interpreting sting values to describe the information or action, which is equivalent functionality of beans on a JSP engine (Appendix D).

**Claims 13-24 and 26-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Voice extensible Markup Langauge (voiceXML) version 1.0 (as cited above), in view of Sorsa (US 6424945, filed Dec 1999), further in view of Dodrill'564 (US 06490564, filed Feb 2000).**

**Regarding independent claim 13**, Vxml does not expressly teach, but Sorsa teaches a voiceXML processor for parsing and rendering a voiceXML document. For example, Sorsa discloses Sorsa discloses (in col 5, lines 62-67), a Voice browser 120 can be implemented as hardware, software, or a combination of both. Voice browser 120 is preferably capable of interpreting a markup language, such as VoiceXML or other similar language used for speech-enabled browsers. The markup language should describe the speech content (i.e., the audio output), voice commands (including their grammar and rules for state transition, which in the current invention is state-

dependent), and voice input items (and actions corresponding to the input items)(col 5, lines 62-67).

Vxml teaches a library comprising reusable voicexml dialog components that are accessible by the voicexml parser, wherein the voicexml document comprises code for invoking a reusable voicexml dialog component and code for configuring the invoked reusable voicexml dialog component using an associated parameter object. The Vxml reference discloses subdialog elements execute with state information which permit the reuse of a common dialog such as prompting a user for credit card information or building libraries of reusable applications (Vxml, section 14.4).

*Vxml does not expressly teach, wherein the step of creating a reusable voiceXML dialog component comprises creating a re-entrant reusable voicexml dialog component that allow reusable voicexml dialog component to be one of initiated, interrupted, inspected or resumed with a partially filled result object or state object.*

However, Vxml does teach creating a reusable library of dialogs shared among many applications with sessions where the user starts to interact with a voice interpreter context (see vxm, section 3, 3.1, 3.2) and voicexml elements for “<initial>,” which declares initial logic upon entry into a (mixed-initiative) form (see vxm, section 4).

Additionally, The Dodrill patent reference discloses development of a voice enabled web application using a browser, where the server parses the existing XML document that defines a voice application operation, inserts selected tag data into fields of a form in order to create or modify voice application operations in the entry fields, where the application server stores the XML document for later execution (col 7, lines 45-67).

Examiner interprets that creation or modification of a voice enabled XML document that is stored for modification is equivalent to the claimed creating because building components is the same thing as creating them.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify vxml to include interpreting a markup language such as voicexml as taught by sorsa, providing the benefit of having an improved system for providing IVR services, where accurate speech recognition is achieved having modest processing capability and memory resources, where the speech recognition uses grammars having limited size (Sorsa, col 2, lines 26-31), further to include initializing voicexml elements upon entry into a mixed-initiative from as taught by Vxml, providing the benefit of bringing the advantages of web-based development and content delivery to interactive voice response applications (voicexml, abstract).

**Regarding claim 14** Vxml teaches reusable voicexml dialog component is invoked using a subdialog elements. Vxml discloses creating reusable components (section 5, Subdialog heading).

**Regarding claim 15**, Vxml teaches parameter ECMAScript. Vxml discloses specifying a block of ECMAScript (col 4, lines 20-21).

**Regarding claim 16**, Vxml teaches default prompts, object-specific resources, constructors that combine default and application specific parameters, and a method for manipulating parameter content, and a combination thereof. Vxml discloses a platform-specific document message to prompt users (section 5, Main Document example) containing resources relevant to object (section 14.5 – table, see archive object) with

attributes have either default or specified fields by the last user input)(section 15, Filled heading and a retrieval of the content (section 12.1, see fetchint).

Combining these elements would have been obvious to one of ordinary skill in the art at the time of the invention because the vxml reference teaches all of the elements providing the benefit of reusable components to take advantages of web-based development and content delivery to interactive voice response application (vxml, Abstract section).

**Regarding claim 17,** Vxml teaches the library of reusable voicexml dialog components that is maintained on a server or maintained locally. Vxml discloses subdialog elements execute with state information which permit the reuse of a common dialog such as prompting a user for credit card information or building libraries of reusable applications (Vxml, section 14.4). By building the library of reusable applications, the system can reuse previously stored dialogs after it has been initially built.

**Regarding claim 18,** Vxml teaches default grammars and audio prompts to support behavior of the reusable voicexml dialog components. Vxml discloses audio clip with a prompt (section 4 table).

**Regarding claim 19,** Vxml teaches library further maintains a reusable voicexml dialog module comprising a standardized set of reusable voicexml dialog components. Vxml discloses a standard session variables are reusable and can are stored in the library (section 9.4).

**Regarding claim 20**, Vxml teaches reusable voicexml module that supports dialog localization for other languages. Vxml discloses attributes of <vxm> include language and local type (section 5).

**Regarding claim 21**, Vxml teaches repository ECMAScript functions. Vxml discloses subdialogs that permit reuse of common dialog including ECMAScript object (section 14.4).

**Regarding claim 22**, Vxml teaches repository for dynamic grammar compilers and audio prompt editors, which can be ported to the voicexml processor platform. Vxml discloses audio clip within a prompt that were created (section 4, table <audio> element).

**Regarding claim 23**, Vxml does not expressly teach, but Sorsa teaches voicexml browser. Vxml discloses a voicexml language used for speech enabled browser (col 5, lines 64-67).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify vxml to include voicexml used for speech enabled browsers as taught by sorsa, providing the benefit of having an improved system for providing IVR services, where accurate speech recognition is achieved having modest processing capability and memory resources, where the speech recognition uses grammars having limited size (Sorsa, col 2, lines 26-31).

**Regarding claim 24**, Vxml does not expressly teach, but Sorsa teaches speech application server provides a speech interface for a multi-modal browser. Sorsa

discloses an audio input devices and voice browser (fig 2, item 102, 204, 120; col 5, lines 15-18).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify vxml to include voicexml used for speech enabled browsers as taught by sorsa, providing the benefit of having an improved system for providing IVR services, where accurate speech recognition is achieved having modest processing capability and memory resources, where the speech recognition uses grammars having limited size (Sorsa, col 2, lines 26-31).

**Regarding claim 26**, Vxml teaches re-entrant objects are used for mixed initiative. Vxml discloses a mixed-initiative conversation (Abstract, 6.2.2, <initial>).

**Regarding claim 27**, Vxml teaches reusable voicexml dialog component comprising an object element for dynamic data access. Vxml discloses a voice browser system that is able to receive speech recognition grammar data dynamically section 2.5).

**Regarding independent claim 28**, Vxml does not expressly teach, but Sorsa teaches receiving and parsing document. Sorsa discloses (in col 5, lines 62-67), a Voice browser 120 can be implemented as hardware, software, or a combination of both. Voice browser 120 is preferably capable of interpreting a markup language, such as VoiceXML or other similar language used for speech-enabled browsers. The markup language should describe the speech content (i.e., the audio output), voice commands (including their grammar and rules for state transition, which in the current invention is state-dependent), and voice input items (and actions corresponding to the input items).

Vxml teaches invoking a reusable voicexml dialog components using a subdialog element; instantiating an associated parameter object for configuring the invoked reusable voicexml document. Vxml discloses subdialogs for invoking new interactions to create a reusable library of dialogs shared among applications (section 3.1, Dialogs and Subdialogs).

Vxml in view of Sorsa does not expressly teach, but Dodrill'564 suggests *Dynamically compiling a grammar for the invoked reusable voicexml dialog component.* Dodrill'564 discloses the XML documents are then stored for execution of the voice application by an application server in the application runtime environment. Hence, web based voice applications can now be developed using open-source XML document development tools such as forms-based document development systems, as opposed to development environments that require compiling applications written in programming languages such as C, C++, PERL, Java, etc.; the examiner interprets execution by the application server in the application runtime environment as equivalent to dynamic compilation of markup language components (col 5, lines 60-67). The examiner interprets Dodrill'564's runtime environment, as opposed to the compiling environment, as equivalent to the dynamically compiling because dynamically compiling is compiling at the moment of execution without a preprocess compilation which is later rendered as an executable. The execution at runtime does not require a preprocess compilation process and allows for execution/interpretation on-the-fly, at the moment of execution. Dodrill'564 abundantly teaches voice application on XML documents which the examiner interprets as equivalent to eh application of the voiceXML dialog. The claimed

invention claims the dynamically compiling a grammar as the positive limitation for the speech application. The examiner further interprets the invoked reusable voicexml dialog component as a functional use and not a positive limitations. Thus, the examiner asserts the broadest reasonable interpretation of dynamically compiling a grammar and asserts the Vxml reference in view of Sorsa and Dodrill'564 to teach the positive limitation of dynamically compiling a grammar. Vxml teaches the use of grammars for the voicexml and the Dorill'564 reference teaches voice enabled xml applications executing during application runtime.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify vxml to include a voice browser interpreting a markup language such as voicexml as taught by sorsa, providing the benefit of having an improved system for providing IVR services, where accurate speech recognition is achieved having modest processing capability and memory resources, where the speech recognition uses grammars having limited size (Sorsa, col 2, lines 26-31), further to include execution of the voice application by an application server in the application run environment as taught by Dodrill'564, providing the benefit of voice enabled web can now be developed by individuals without the necessity of programming language experience (Dodrill'564, col 6, lines 1-10) .

**Regarding claim 29**, Vxml teaches instantiating an associated parameter object comprising using ECMAScript. Vxml discloses specifying a block of ECMAScript (col 4, lines 20-21).

**Regarding claim 30,** Vxml teaches maintaining a repository comprising a library of reusable voicexml dialog component and associated parameter objects. Vxml discloses a reusable library of dialogs for voicexml documents (section 3.1; section 4, table with <parm> elements).

**Regarding claim 31,** vxml teaches maintaining a repository of default grammars and audio prompts. Vxml discloses default scope of the form's grammars with prompts that are revisited, and in order to revisit the prompts, they must be stored (section 6, forms table)(section 4, Voicexml <reprompt>).

**Regarding independent claim 32,** Vxml does not expressly teach, but Sorsa teaches receiving and parsing document. Sorsa discloses (in col 5, lines 62-67), a Voice browser 120 can be implemented as hardware, software, or a combination of both. Voice browser 120 is preferably capable of interpreting a markup language, such as VoiceXML or other similar language used for speech-enabled browsers. The markup language should describe the speech content (i.e., the audio output), voice commands (including their grammar and rules for state transition, which in the current invention is state-dependent), and voice input items (and actions corresponding to the input items).

Vxml teaches invoking a reusable voicexml dialog components using a subdialog element; instantiating an associated parameter object for configuring the invoked reusable voicexml document. Vxml discloses subdialogs for invoking new interactions to create a reusable library of dialogs shared among applications (section 3.1, Dialogs and Subdialogs).

Vxml in view of Sorsa does not expressly teach, but Dodrill'564 suggests

*Dynamically compiling a grammar for the invoked reusable voicexml dialog component.*

Dodrill'564 discloses the XML documents are then stored for execution of the voice application by an application server in the application runtime environment. Hence, web based voice applications can now be developed using open-source XML document development tools such as forms-based document development systems, as opposed to development environments that require compiling applications written in programming languages such as C, C++, PERL, Java, etc.; the examiner interprets execution by the application server in the application runtime environment as equivalent to dynamic compilation of markup language components (col 5, lines 60-67). The examiner interprets Dodrill'564's runtime environment, as opposed to the compiling environment, as equivalent to the dynamically compiling because dynamically compiling is compiling at the moment of execution without a preprocess compilation which is later rendered as an executable. The execution at runtime does not require a preprocess compilation process and allows for execution/interpretation on-the-fly, at the moment of execution.

Dodrill'564 abundantly teaches voice application on XML documents which the examiner interprets as equivalent to eh application of the voiceXML dialog. The claimed invention claims the dynamically compiling a grammar as the positive limitation for the speech application. The examiner further interprets the invoked reusable voicexml dialog component as a functional use and not a positive limitations. Thus, the examiner asserts the broadest reasonable interpretation of dynamically compiling a grammar and asserts the Vxml reference in view of Sorsa and Dodrill'564 to teach the positive limitation of dynamically compiling a grammar. Vxml teaches the use of grammars for

the voicexml and the Dorill'564 reference teaches voice enabled xml applications executing during application runtime.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify vxml to include a voice browser interpreting a markup language such as voicexml as taught by sorsa, providing the benefit of having an improved system for providing IVR services, where accurate speech recognition is achieved having modest processing capability and memory resources, where the speech recognition uses grammars having limited size (Sorsa, col 2, lines 26-31), further to include execution of the voice application by an application server in the application run environment as taught by Dodrill'564, providing the benefit of voice enabled web can now be developed by individuals without the necessity of programming language experience (Dodrill'564, col 6, lines 1-10).

#### **(10) Response to Argument**

Regarding independent claim 1, Appellant argues that the combination of references (the Vxml non-patent literature and Dodrill) do not teach the claim limitations of creating a re-entrant reusable voicexml dialog component that allows reusable voicexml dialog components to be one of initiated, inspected or resumed with a partially filled result object or state object (see Brief, pages 9-11). The examiner disagrees. The examiner characterizes this limitation as reusable voicexml components that are created (in view of appellant's specification section, pages 50-56). To teach this limitation, the Vxml reference teaches creating a reusable library of dialogs shared among many applications (Vxml, sections 3, 3.1 and 3.2). Additionally, the Vxml

reference discloses subdialog elements execute with state information which permit the reuse of a common dialog such as prompting a user for credit card information or building libraries of reusable applications (Vxml, section 14.4). By building the library of reusable applications, the system can reuse previously stored dialogs after it has been initially built. The examiner interprets the disclosed "building" as equivalent to the claimed "creating" because building requires creating. Additionally, the Dodrill patent reference discloses development of a voice enabled web application using a browser, where the server parses the existing XML document that defines a voice application operation, inserts selected tag data into fields of a form in order to create or modify voice application operations in the entry fields, where the application server stores the XML document for later execution (col 7, lines 45-67). Examiner interprets that creation or modification of a voice enabled XML document that is stored for modification is equivalent to the claimed creating because building components is the same thing as creating them. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Vxml with Dodrill to include development of a voice enabled web application using a browser, where the server parses the existing XML document that defines a voice application operation, inserts selected tag data into fields of a form in order to create or modify voice application operations in the entry fields, where the application server stores the XML document for later execution as taught by Dodrill, providing the benefit of a non-conventional application for voice enabled web applications using a browser, enabling users to input or modify application parameters for the XML document in the form (col 3, lines 47- col 4, line 1).

Regarding claim 33, Appellant argues that Vxml in view of Dodrill does not suggest a first database comprising one or more server-side reusable voicexml dialog components that are accessible by the voicexml page generation engine for generating an intermediate voicexml page (see Brief, pages 11-12). The examiner disagrees. The examiner characterizes this limitation as a data store that allows voicexml related data storage for reuse. To teach this characterization, Dodrill's discloses where a user can modify an application user the HTML forms and then submit the HTMI forms to the application server which generates corresponding XML document stored for later execution (col 8, lines 11-28, col 7, lines 40-67; fig 2). The examiner interprets Dodrill's storing of the XML document for later execution as equivalent to the database comprising reusable voicexml dialog components because the storing xml documents with voice enabled data must be stored in a data store for performing the storing step. Dodrill's voice enabled web application uses XML and stores the date because the server parses the existing XML document that defines a voice application operation, inserts selected tag data into fields of a form in order to create or modify voice application operations in the entry fields, where the application server stores the XML document for later execution (col 7, lines 45-67). Additionally, the reference Vxml discloses Vxml subdialogs that permit the reuse of common dialog or build libraries of reusable applications for VXML applications. Since the vxml subdialogs are stored for reuse, they must be stored in some form of a data store. The combination of the two references, Vxml and Dodrill abundantly discloses a data store that allows for storage of voicexmzl related data for reuse.

Regarding claim 34, Appellant argues that the Vxml reference does not teach reusable voicexml dialog components using beans and wherein the voicexml page generation engine comprises a Java Server Page engine (page 13). The examiner disagrees. Vxml discloses the use of Java which can be used with voiceXML grammar elements (Vxml, Appendix D) which is equivalent to the claimed limitation because Java was well known in the art at the time of the invention and using Java beans was also well known in the art at the time of the invention to develop web server based reusable components for web applications to server data in forms, and this is evidenced by the Java Speech Grammar Format that defines a grammar for interpreting sting values to describe the information or action, which is equivalent functionality of beans on a JSP engine.

Regarding claim 13, Appellant argues that Vxml in view of Sorsa does not teach wherein the reusable voicexml dialog components comprising one or more reentrant reusable voicexml dialog component that allow reusable voicexml dialog components to be one of initiated, interrupted, inspected or resumed with a partially filled result object or state object. The examiner disagrees. The examiner characterizes this limitation as reusable voicexml components that are created (in view of appellant's specification section, pages 50-56). To teach this limitation that the Appellant argues is lacking from the combination of references, the Vxml reference discloses creating a reusable library of dialogs shared among many applications (Vxml, sections 3, 3.1 and 3.2). Additionally, the Vxml reference discloses subdialog elements that execute with state information which permit the reuse of a common dialog such as prompting a user for

credit card information or building libraries of reusable applications (Vxml, section 14.4). By building the library of reusable applications, the system can reuse previously stored dialogs after it has been initially built. The examiner interprets the disclosed “building” as equivalent to the claimed “creating” because building requires creating. The examiner relies on the Sorsa reference to teach a voicexml processor for parsing and rendering a voicexml document. For example, Sorsa discloses (in col 5, lines 62-67), a Voice browser 120 can be implemented as hardware, software, or a combination of both. Voice browser 120 is preferably capable of interpreting a markup language, such as VoiceXML or other similar language used for speech-enabled browsers. The markup language should describe the speech content (i.e., the audio output), voice commands (including their grammar and rules for state transition, which in the current invention is state-dependent), and voice input items (and actions corresponding to the input items). The Appellant argues that the examiner does not assert how Sorsa discloses reusable voicexml dialog components that are “re-entrant”. The examiner disagrees with Appellant’s characterization because the examiner does not assert the Sorsa reference to teach “re-entrant”. For the “re-entrant” limitation, the examiner asserts the Vxml nonpatent literature reference (see immediately above arguments). The examiner asserted the Sorsa reference to show that voice enabled browsers with processors for voicexml were well known at the time of the invention (see Sorsa, col 5, lines 62-67).

Regarding claims 28 and 32, Appellant argues were not obvious in view of the Vxml, Sorsa and the Dodrill’564 references (see Brief, pages 14-15). Specifically, the

references do not suggest dynamically compiling a grammar for the invoked reusable voicexml dialog component. The examiner disagrees. Dodrill'564 discloses the XML documents are then stored for execution of the voice application by an application server in the application runtime environment. Hence, web based voice applications can now be developed using open-source XML document development tools such as forms-based document development systems, as opposed to development environments that require compiling applications written in programming languages such as C, C++, PERL, Java, etc.; the examiner interprets execution by the application server in the application runtime environment as equivalent to dynamic compilation of markup language components (col 5, lines 60-67). The examiner interprets Dodrill'564's runtime environment, as opposed to the compiling environment, as equivalent to the dynamically compiling because dynamically compiling is compiling at the moment of execution without a preprocess compilation which is later rendered as an executable. The execution at runtime does not require a preprocess compilation process and allows for execution/interpretation on-the-fly, at the moment of execution. Dodrill'564 abundantly teaches voice application on XML documents which the examiner interprets as equivalent to the application of the voiceXML dialog. The claimed invention claims the dynamically compiling a grammar as the positive limitation for the speech application. The examiner further interprets the invoked reusable voicexml dialog component as a functional use and not a positive limitation. Thus, the examiner asserts the broadest reasonable interpretation of dynamically compiling a grammar and asserts the Vxml reference in view of Sorsa and Dodrill'564 to teach the positive

limitation of dynamically compiling a grammar. Vxml teaches the use of grammars for the voicexml and the Dorill'564 reference teaches voice enabled xml applications executing during application runtime.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

*GS 10/12/06*  
Gautam Sain, Patent Examiner, 10/12/2006

Conferees:

*Heather Herndon*  
Heather Herndon, Supervisory Patent Examiner

*Heather Herndon*  
HEATHER R. HERNDON  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100

Steve Hong, Supervisory Patent Examiner

*Stephen Hong*  
STEPHEN HONG  
SUPERVISORY PATENT EXAMINER